ID5001W UoP Assignment

Course Instructor : Arun Rajkumar

WHAT SHOULD YOU SUBMIT? You should submit a zip file titled 'Solutions roll-number.zip' where rollnumber is your roll number. Your assignment will NOT be graded if it does not contain all of the following:

- A text file titled 'Participant detail.txt' with your name and roll number.
- A PDF file which includes explanations regarding each of the solution as required in the question. Title this file as 'Report.pdf'
- Source code for all the programs that you write for the assignment clearly named.

GUIDELINES: Keep the below points in mind before submission.

- Plagiarism of any kind is unacceptable. These include copying text or code from any online sources. These will lead to disciplinary actions according to institute guidelines.
- Any graph that you plot is unacceptable for grading unless it labels the x-axis and y-axis clearly.
- Don't be vague in your explanations. The clearer your answer is, the more chance it will be scored higher.
- For programming components, your submission must include:
  - Well-commented Python code.
  - Detailed epoch-wise training results and discussion.
  - A clearly labeled visualization plot.
  - Predictions and interpretation of classifier behavior.

1 Perceptron

Suppose we have a dataset with four points in a 2D feature space:

$$\begin{aligned} \mathbf{x}_1 &= (2,2), \quad y_1 = +1 \\ \mathbf{x}_2 &= (4,4), \quad y_2 = +1 \\ \mathbf{x}_3 &= (1,-1), \quad y_3 = -1 \\ \mathbf{x}_4 &= (-3,-3), \quad y_4 = -1 \end{aligned}$$

Consider the standard perceptron algorithm with initialization as $\mathbf{w}^{(0)} = (0,0)$ and bias $b^{(0)} = 0$. The update rule is given by:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i, \quad b^{(t+1)} = b^{(t)} + y_i$$

where $(\mathbf{x}_i, y_i)$ is a misclassified example at iteration t.

(a) Perform two full epochs of the perceptron training algorithm (cycling through all points in the listed order). Show the final weight vector $\mathbf{w}$, bias b, and classification outcomes after each individual update.

(b) After two epochs, is the data linearly separable? Give mathematical justification by explicitly verifying if your final perceptron hyperplane separates the classes correctly.

(c) Compute the margins γ for each point with respect to the final perceptron hyperplane (from part b) using:

$$\gamma = \min_i \frac{y_i(\mathbf{w} \cdot \mathbf{x}_i + b)}{\|\mathbf{w}\|}$$

Which point has the minimum margin?

Programming Question

Consider the following two-dimensional dataset:

Point	x_1	x_2	y
1	0	0	-1
2	1	1	-1
3	1	0	$+1$
4	0	1	$+1$
5	0.5	0.5	-1
6	2	2	-1
7	2	0	$+1$
8	0	2	$+1$

(a) Implement the perceptron algorithm from scratch in Python. Use the perceptron update rules as in previous section. Run it for a maximum of 15 epochs or until convergence. Clearly provide epoch-wise training results, for $\mathbf{w}$ and bias b, and number of mistakes per epoch

(b) Now transform the original dataset using the following non-linear mapping:

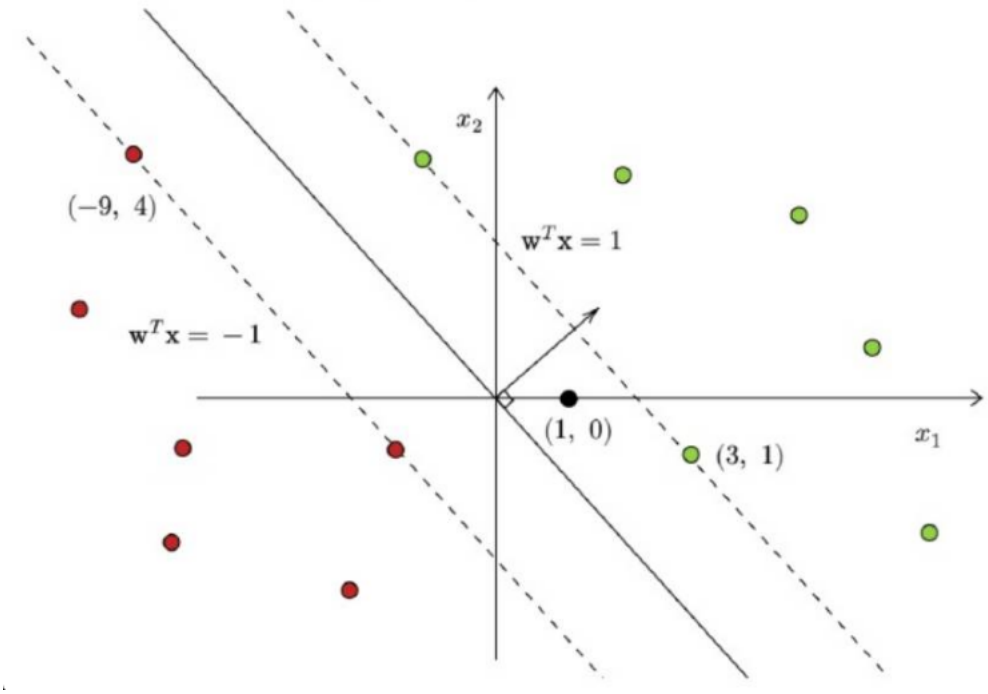
$$\phi(x_1, x_2) = (x_1, x_2, x_1^2 + x_2^2)$$

Train the perceptron again using this transformed dataset. Provide epoch-wise training results clearly, and explicitly state whether the algorithm converges in this new feature space.

(c) Plot the original two-dimensional dataset, clearly differentiating the two classes. Using your learned perceptron (trained on transformed features), plot the corresponding **decision boundary** in the original two-dimensional space.

2 SVM

1. Consider a hard-margin SVM trained on a dataset in $\mathbb{R}^2$ for a binary classification task. Red and green points belong to the training dataset. Red points belong to class -1 and green points belong to class +1. The black point is a test data-point. The dotted lines are the supporting hyperplanes for the SVM. Note: We don't have access to the test data-point during training; it is given to us after the model has been learned on the training dataset.



(a) What is the equation of the decision boundary?

(b) What is the width of separation between supporting hyperplanes?

Programming Question

2. Find the hinge loss for this soft-margin SVM classifier on the dataset that is given in the table. The weight vector and bias are as follows:

$$\mathbf{w} = [1, -1]^T, \quad b = 1$$

The coefficient C can be assumed to be 1.

x_1	x_2	y
1	4	-1
-1	2	-1
0	0	-1
1	2	-1
1	3	1
1	0	1
2	1	1
2	3	1

Note that you just need to report the hinge loss. Do not compute margin loss, which involves only the term $\mathbf{w}$. Also, note that the hinge loss does **not** have a factor of 0.5 before it. Enter the closest integer as your answer.

3 K-Means

Use the K-means algorithm and Euclidean distance to cluster the following 8 examples into 3 clusters:
A1 = (2, 10), A2 = (2, 5), A3 = (8, 4), A4 = (5, 8), A5 = (7, 5),
A6 = (6, 4), A7 = (1, 2), A8 = (4, 9).

The distance matrix based on the Euclidean distance is given below:

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

Suppose that the initial seeds (centers of each cluster) are A1, A4, and A7. Run the K-means algorithm for one epoch only. At the end of this epoch, answer the following:

(a) Determine the new clusters (i.e., which examples belong to each cluster).

(b) Compute the centers of the new clusters.

(c) Draw a 10×10 space with all the 8 points and show the clusters after the first epoch along with their new centroids.

(d) Estimate how many more iterations are needed for convergence. Draw the result for each epoch.

(e) Consider the points $z_1, z_2, \ldots, z_m$, where $m \geq 1$, and for $i \in \{1, 2, \ldots, m\}$, $z_i \in \mathbb{R}^d$. Let $\overline{z} = \frac{1}{m} \sum_{i=1}^{m} z_i$ be the mean of these points, and let $z \in \mathbb{R}^d$ be an arbitrary point in the same (d-dimensional) space. Then show that

$$\sum_{i=1}^{m} \|z_i - z\|^2 \geq \sum_{i=1}^{m} \|z_i - \overline{z}\|^2.$$

(f) Consider the K-means algorithm with the following *Greedy initialization* method for selecting k initial centroids from a dataset $X = \{x_1, x_2, \ldots, x_n\}$ in $\mathbb{R}^d$:

(i) Choose the first center c_1 uniformly at random from X.

(ii) For $i = 2, 3, \ldots, k$, choose the next center $c_i = \arg\max_{x \in X} D(x)$, where $D(x) = \min_{c \in C} \|x - c\|$ and $C = \{c_1, c_2, \ldots, c_{i-1}\}$ is the current set of centers.

After initialization, the algorithm proceeds as standard K-means:

$$C_t = \{C_{t,1}, C_{t,2}, \ldots, C_{t,k}\} \text{ is the clustering at iteration } t,$$

$$\mu_t = \{\mu_{t,1}, \mu_{t,2}, \ldots, \mu_{t,k}\} \text{ are the centroids, where } \mu_{t,j} = \frac{1}{|C_{t,j}|} \sum_{x \in C_{t,j}} x,$$

$$\text{SSE}(C_t, \mu_t) = \sum_{j=1}^{k} \sum_{x \in C_{t,j}} \|x - \mu_{t,j}\|^2.$$

Using results from (a), show that the algorithm still converges (i.e.,

$$\text{SSE}(C_{t+1}, \mu_{t+1}) \leq \text{SSE}(C_t, \mu_t)$$

with strict inequality when $C_{t+1} \neq C_t$). Then, argue how the Greedy initialization affects convergence compared to random initialization, and discuss if k^n possible clusterings guarantee termination.

4 GMM

1. We wish to fit a GMM with $K = 2$ for a dataset having 4 points. At the beginning of the tth time step of the EM algorithm, we have $\theta^{(t)}$ as follows: $\pi_1 = 0.3$, $\pi_2 = 0.7$ $\mu_1 = 2$, $\sigma_1^2 = 1$ $\mu_2 = 3$, $\sigma_2^2 = 1$

   The density of the points given a particular mixture is given to you for all four points. f is the density of a Gaussian.

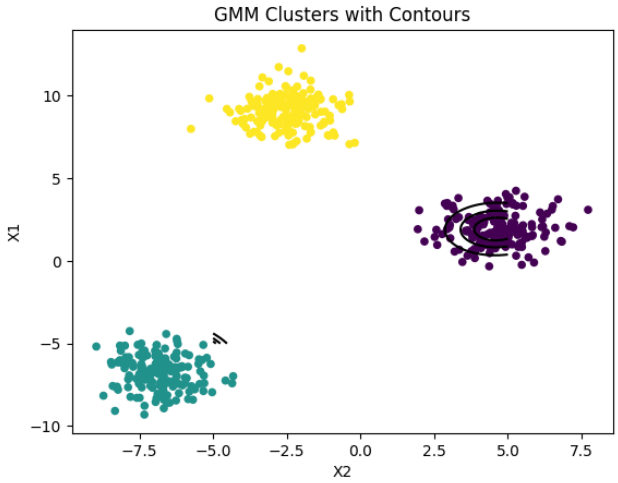


Figure 1: Example visualization

x_i	$f(x_i\|z_i = 1)$	$f(x_i\|z_i = 2)$
1	0.242	0.054
2	0.399	0.242
3	0.242	0.399
4	0.054	0.242

(a) What is the value of λ_k^i for $i = 1$ and $k = 2$ after the E-step?

(b) If we pause the algorithm at this stage (after the E-step) and use the λ_k^i values to do a hard-clustering, what would be the cluster assignment? We use the following rule to come up with cluster assignments. $z_i = argmax_k \lambda_k^i$. Give the answer as a vector Z

Programming Question

Lets code the GMM model from scratch. Use only numpy library, make_blobs from sklearn.datasets and multivariate_normal from scipy.stats and implement the GMM algorithm.

1. Create a 2D dataset with the following parameters (n_samples=2000, centers=3, cluster_std=1.0)
2. Split the dataset randomly into train (80%) and test set (20%)
3. Fit the GMM model with 3 cluster centers on the training dataset
4. Predict the test dataset class label using the fitted model
5. Visualize the test dataset with color coding using predicted label